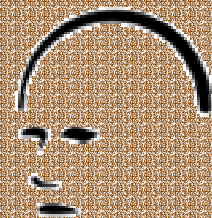


Application of Formal Ontology to Database Schema Alignment - an Outline

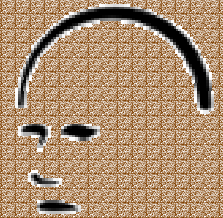
Bill Andersen
Ontology Works, Inc.

Schema alignment?



- UML-based database schemas are impoverished
- OWL-DL-based ontologies suffer the same problem
- What is needed to effect alignment of databases built on these formalisms?
- Depends on what we mean by alignment in the context of federated query

Oracle Sales DB



Employee

Id	Name	Dept

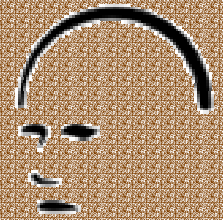
Does 'Employee' correspond to a type?

Are the employee ID numbers given by the company that owns the database or are they surrogate keys?

Employees of what company? It's not in the schema.

And the IDs are useless for combining information about employees across databases from different companies

Oracle Sales DB



Product

Id	Name

Are products individual objects like employees?

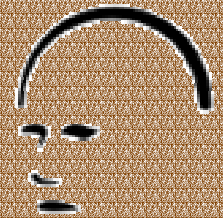
Are product names comparable across DBs?

Order

Id	Prod	Qty

What are orders?

Oracle Sales DB



Department

Id	Name	Mgr

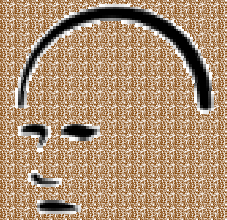
Does 'Department' correspond to a type?

Is an ID of a 'Department' like an ID of an employee?

Departments of what company? It's not in the schema.

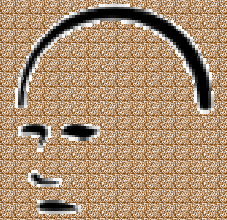
Does combining information about Departments across companies even make sense?

Heuristics aren't enough



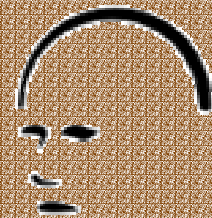
- Lexical matching of schema element names doesn't answer these questions
- Structural heuristics that consider statistically significant clusters of matches don't provide the answers either
- What could?

Formal Ontology



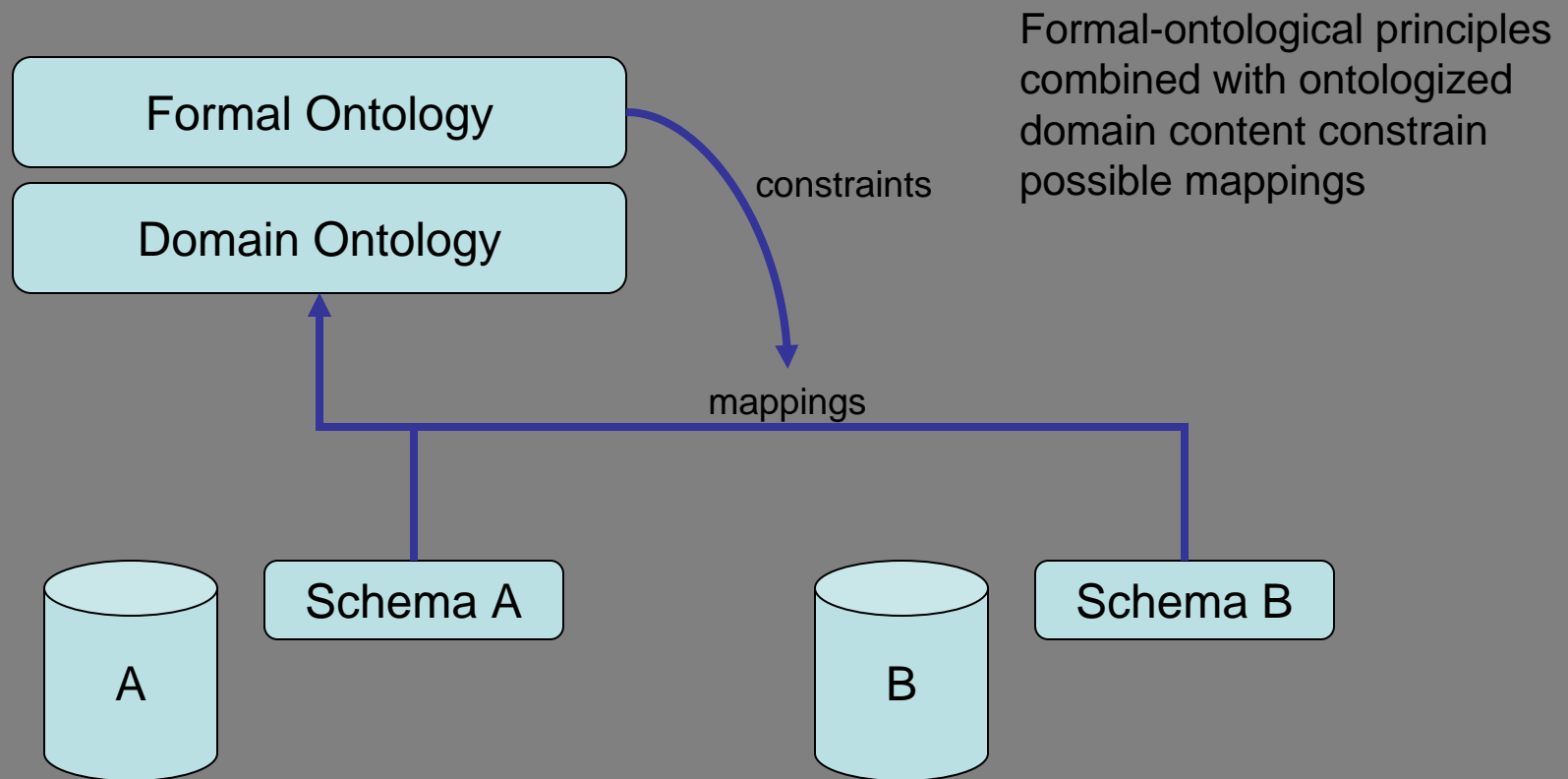
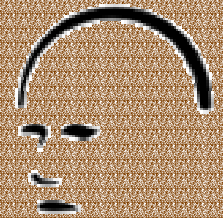
- Concerned with the most general structures of (domain-independent) reality
 - Identity
 - Mereology
 - Dependence
 - Modality and change
- Here we can find some tools that provide the additional information needed to answer our questions

This stuff can't be relevant, can it?

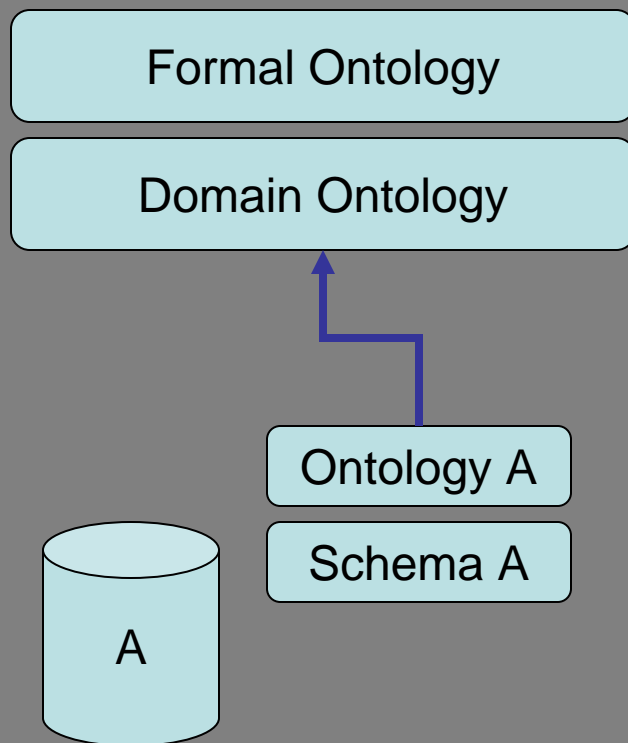
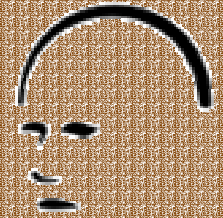


- Dependence
 - Departments and employees are dependent entities ... dependent upon organizations
- Mereology
 - Departments are ultimately part of organizations that are not dependent
- Identity
 - Cross-DB individuation of employees and can't come from employee IDs
- Modality
 - Employeehood is a contingent matter

Ontology-based federation



A sketch of a process



Map Schema A elements to Domain Ontology

Construct elaboration Ontology A based on dependence relations

Construct views that disambiguate information in Database A

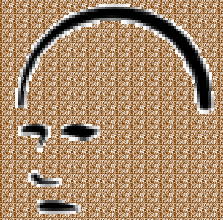
Identify criteria point to where inter-DB concordances are required

Requirements

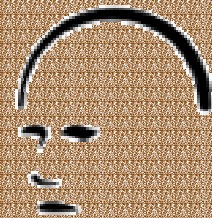


- Ontology language must support limited second-order logic (e.g., SCL)
 - Full power of logic not needed once mappings established (or else we're in trouble)
 - OWL-Full may be sufficient
- Lambda abstraction may be necessary
- Suitable formal ontological content is required (e.g., BFO, DOLCE)

Summary



- Formal-ontological notions provide needed information to control search for mappings
 - Constraints augment heuristic techniques
- Formal-ontological notions fill in missing information
 - Improving accuracy of cross-database query
- Formal-ontological basis provides neutral model for integration of arbitrarily many databases
 - Avoids ontological short-cuts



Questions?